TIPOLOGÍA Y CICLO DE VIDA DE LOS DATOS
2017-2018 SEMESTRE 1 AULA 1

PRÁCTICA 1

CONSULTORA: LAIA SUBIRATS MATÉ
AUTOR: JOSÉ IGNACIO BENGOECHEA ISASA

TÍTULO Y SUBTÍTULO DEL DATASET

El título y subtitulo del proyecto creado en Github, asociado al código, y del dataset es el siguiente:

NBA and WNBA salary gap.

Stats of the NBA and WNBA players related to the gender pay gap.

En español:

Brecha salarial de la NBA y la WNBA.

Estadísticas de los jugadores de la NBA y de las jugadoras de la WNBA relacionadas con la brecha salarial.

El Github con el código y el dataset está en la siguiente dirección:

https://github.com/Bengis/nba-wnba-salary-gap

IMAGEN DEL DATASET

Agregad una imagen que identifique vuestro dataset visualmente

El logo o imagen se inspira en el logo original de la NBA. Este logo realizado en 1969 nos muestra la figura en blanco de un jugador, Jerry West, y dos colores que representan a la conferencia este y oeste de la NBA.

Figura 1.Logo de la NBA. Fuente: Nba.com.

En mi caso he creado un logo base similar en el que la figura base es el jugador español Ricky Rubio. Al ser un logo asociados a las estadísticas le he añadido estadísticas de juego reales.

Figura 2. Logo de Nba Stats. Fuente propia.

CONTEXTO

¿Cuál es la materia del conjunto de datos?

Los datos minados corresponden a estadísticas de baloncesto. La NBA ha cuidado mucho el almacenamiento de las estadísticas asociadas a sus partidos, ya que entienden que son los indicadores que permiten establecer cuáles son los mejores jugadores o lo que más impacto pueden tener sobre un equipo.

El uso de análisis de datos en la NBA se ha incrementado gracias al éxito deportivo de los *Golden State Warriors,* equipo alojado en la ciudad de San Francisco, que ha contado con la ayuda de la tecnología para poder determinar que jugadores jóvenes de los que ofrecía el mercado podían tener una mayor mejora. Se puede consultar información sobre este hecho en este enlace:

https://the-cauldron.com/understanding-the-data-the-golden-state-warriors-and-the-role-of-analytics-37c1b387c7b1

Este tipo de análisis avanzado de estadística trata de explotar las ineficiencias no usadas en las disciplinas deportivas. En el caso de los *Golden State Warriors* se potencio la búsqueda de jugadores jóvenes que fueron efectivos desde la línea de 3 puntos.

CONTENIDO

¿Qué campos incluye?

El dataset muestra medias estadísticas totales de todos los jugadores y jugadoras de la NBA y de la WNBA, por temporada. Trata de mostrarnos la brecha salarial que existe entre los jugadores y las jugadoras. Para ello se realizan los siguientes ajustes:

- Los valores salariales de las jugadoras no han sido publicados por la WNBA. Por ello, se determina que todas las jugadoras ganan el máximo salarial que es posible, el cual se corresponde con la cantidad de 105.000 $.
- Existen valores estadísticos reales, como puntos o rebotes y valores estadísticos calculados en función del salario, como salario/puntos o salario/rebotes. En caso de que el jugador o jugadora no tenga puntos o rebotes el resultado será la cadena de texto “N/A”, por “not available”, ya que el resultado generaría una excepción al dividir por cero.

El dataset incluye los siguientes campos:

- *Player*. Jugador/Jugadora. Es el campo del nombre del jugador.
- *Games*. Partidos. Es el total de partidos que disputaron los jugadores durante la última temporada. En la NBA el máximo es 82 y en la WNBA el máximo es 32. Obviamente esto es favorable para los jugadores ya que al jugar más partidos tienen más puntos, y por lo tanto la brecha salarial se debería reducir. Sin embargo, la brecha salarial es tan grande que el salario por estadística de las chicas es varios ordenes de magnitud inferior.
- *Minutes*. Minutos totales jugados. La suma de todos los minutos jugados en la liga durante la última temporada.
- *Points*. Puntos totales realizados. La suma de todos los puntos realizados en la liga durante la última temporada por ese jugador/a.
- *Rebounds*. Rebotes totales realizados. La suma de todos los rebotes realizados en la liga durante la última temporada por ese jugador/a.
- *Asssits*. Asistencias totales realizadas. La suma de todas las asistencias realizadas en la liga durante la última temporada por ese jugador/a.
- *Steals*. Robos totales realizados. La suma de todos los robos realizados en la liga durante la última temporada por ese jugador/a.
- *Blocks*. Tapones totales realizados. La suma de todos los tapones realizados en la liga durante la última temporada por ese jugador/a.
- *Salary.* Salario del jugador/a. En dólares. En el caso de que sea una jugadora ese salario no estaba disponible por lo que se determina que es el máximo posible de 105.000 $. Incluso poniendo el máximo tenemos que existe una brecha de dos órdenes de magnitud en varios atributos.
- *Salary/Point.* Salario/puntos del jugador/a. Nos indica la cantidad de dólares que supone ese jugador o jugadora por cada punto anotado.
- *Salary/Rebound.* Salario/rebotes del jugador/a. Nos indica la cantidad de dólares que supone ese jugador o jugadora por cada rebote realizado.
- *Salary/Assist.* Salario/asistencias del jugador/a. Nos indica la cantidad de dólares que supone ese jugador o jugadora por cada asistencia realizada.
- *Salary/Steal.* Salario/robos del jugador/a. Nos indica la cantidad de dólares que supone ese jugador o jugadora por cada robo realizado.
- *Salary/Block.* Salario/tapones del jugador/a. Nos indica la cantidad de dólares que supone ese jugador o jugadora por cada tapón realizado.

¿Cuál es el periodo de tiempo de los datos y cómo se ha recogido?

El dataset recoge datos de la última temporada completa jugada. En el caso de la NBA esa fue la 2016-2017, que comenzó en octubre de 2016 y acabo en abril del 2017. En el caso de la WNBA esa temporada fue la 2017, que comenzó en mayo de 2017 y acabo en septiembre de 2017.

Los datos de las estadísticas de la NBA se han recogido mediante un proceso de *web scrapping*. Se ha utilizado *Python* y como sistema de *scrapping* se ha usado el paquete *lxml*, el cual nos permite realizar búsquedas mediante *XPath.*

AGRADECIMIENTOS

¿Quién es propietario del conjunto de datos?

El origen real de las estadísticas de la NBA proviene de la misma NBA que ofrece estos datasets desde su propia página. La NBA permite en sus condiciones de uso usar las estadísticas si se hacen sin interés comercial y si se indica en las mismas que el origen es la propia página de la NBA.

Sin embargo, existe un debate en este asunto. En ocasiones, las ligas deportivas han tratado de demandar los juegos que usan estadísticas deportivas alegando que las estadísticas han sido recolectadas por ellos y por lo tanto estos juegos deben pagar alguna licencia, pero estas causas de copyright no han sido fructuosas para las ligas, ya que en Estados Unidos los hechos no tienen copyright y las estadísticas no avanzadas se consideran hechos. Se puede ver más información en el siguiente enlace:

https://www.americanbar.org/newsletter/publications/law_trends_news_practice_area_e_newsletter_home/fantasymeetsreality.html

Las páginas oficiales de la NBA y la WNBA están basadas en tecnología Angular, por lo que se descarga una aplicación al cliente que establece la comunicación con el servidor y que se encarga de mostrar los datos en el navegador. Para poder hacer *scrapping* a estas páginas es necesario usar una librería llamada *selenium* que permite conectarse a un navegador, ya sea *PhantomJs* o *Firefox*, y descargarse los datos. En la práctica esto obliga a que la aplicación abra un navegador. Esta opción, aunque es posible se descartó porque resultaba molesto ver la apertura del navegador.

Por otro lado, existe la opción de utilizar los *endpoints*, estos son conexiones directas a datos en JSON que se pueden consultar desde la página. Sin embargo, me pareció una solución muy fácil, que no requiere utilizar ni *BeatifulSoap* ni *lxml* por lo que quedó descartado. Un ejemplo de este endpoint se puede ver en el siguiente enlace, donde se pueden consultar los datos de los partidos del 14 de Febrero de 2015:

http://stats.nba.com/stats/scoreboard/?GameDate=02/14/2015&LeagueID=00&DayOffset=0

Por ello, busque otra fuente de datos. Hay múltiples páginas que ofrecen estadísticas de la NBA. He usado las siguientes páginas:

- Para las estadísticas de la NBA. Rotowire.com. Rotowire es una aplicación web que permite crear equipos de baloncesto basados en estadísticas reales.

Sus condiciones de uso no nos indican nada acerca del acceso mediante *scrapping* por lo que se entiende que está permitido acceder de forma automática, siempre que sea sin finalidad comercial. Los datos a los que accedemos son datos cuyo origen está en la NBA, las estadísticas son hechos, por lo que no tienen copyright. La página a la que accedemos mediante un comando POST es la siguiente:

https://www.rotowire.com/basketball/player-stats.php

- Para las estadísticas de la WNBA. Rotowire.com. Al igual que en el caso de la NBA tenemos las estadísticas de las jugadoras disponibles. El enlace al que accedemos es el siguiente:

  https://www.rotowire.com/wnba/player-stats-byseason.php

- Para los salarios de los jugadores de la NBA. Spotrac.com. Es una página similar a rotowire.com, donde se crean equipos de jugadores basados en estadísticas reales. En esta página podemos consultar el salario de la última temporada completa de cada jugador. En sus condiciones permiten el uso de datos siempre que no sea con motivos comerciales.

  http://www.spotrac.com/nba/rankings

INSPIRACIÓN

¿Por qué es interesante este conjunto de datos?

Este dataset es interesante tanto para aficionados a la NBA y a la WNBA como para personas que quieran obtener datos acerca de la brecha salarial en los deportes profesionales. Si se consultan los datos y se realizan visualizaciones de cuáles son los jugadores y jugadoras más económicas por asistencia, veremos que mientras los chicos tienen un coste de 10.000$ por asistencia las chicas tienen un coste de 100$ por asistencia.

Es decir, hablamos de dos órdenes de magnitud. Eso quiere decir que el coste por productividad de un jugador puede representar el coste por productividad de 100 jugadoras.

¿Qué preguntas le gustaría responder la comunidad?

Respondiendo al enunciado, la pregunta obvia que responde es: ¿Cuál es la brecha salarial que existe entre jugadores de la NBA y jugadoras de la WNBA por su productividad?

Sin embargo, el estudio de estos datos nos puede dar lugar a preguntas mucho más elaboradas, ya que al final nos llevara a preguntarnos porque dos ligas que juegan el mismo deporte, con los mejores jugadores y jugadoras, donde vemos marcadores y estadísticos parecidos, tiene tal brecha salarial.

Esto puede llevar a plantearnos cómo es posible que la NBA sea tan exitosa. El salario de los jugadores viene de la mitad del dinero ingresado por la NBA cada año. Que los jugadores ganen tanto dinero es consecuencia de que la NBA gana mucho más dinero.

LICENCIA

Seleccionad una de estas licencias y decid porqué la habéis seleccionado:

- **Released Under CC0: Public Domain License**
- **Released Under CC BY-NC-SA 4.0 License**
- **Released Under CC BY-SA 4.0 License**
- **Database released under Open Database License, individual contents under Database Contents License**
- **Other (specified above)**
- **Unknown License**

Para el dataset se ha escogido la licencia *CC BY-NC-SA 4.0 License*. Esta es una licencia que permite compartir y adaptar los datos, pero no permite el uso comercial de los mismos.

El motivo de usarla ha sido porque es una licencia que permite que los datos sean compartidos y modificados, pero no con fines comerciales. Es un tipo de licencia que pienso que fomenta el uso de los datos, que al fin y al cabo es uno de los motivos por los que se crean.

Para el código se ha usado la licencia MIT, esta licencia permite usar el código, modificarlo, publicarlo e incluso comercializarlo. Las licencias como MIT o GPL son licencias especializadas en código que se adaptan mejor que las licencias de CC u ODbL.

El motivo de usar la licencia MIT es que es una licencia *open source* que permite que sus derivados publicados no tengan que ser *open source*, a diferencia de GPL, la cual si obliga a que lo sean en caso de publicación. Al igual que con los datos trato de ofrecer la opción más permisiva.

CÓDIGO

Hay que adjuntar el código con el que habéis generado el dataset, preferiblemente con R o Python, que os ha ayudado a generar el dataset.

El código está disponible en GitHub en la siguiente dirección:

https://github.com/Bengis/nba-wnba-salary-gap/blob/master/code/nba-gap.py

A continuación, comentamos las partes más relevantes del mismo. El siguiente código se corresponde con el código principal del programa. Primero hacemos el scrapping de datos de la NBA, del salario de la NBA y su exportación. Luego hacemos el scrapping de datos de la WNBA, el cálculo de salarios, y su exportación.

```
linelen=14
urlWNBA="https://www.rotowire.com/wnba/player-stats-byseason.php"
urlNBA="https://www.rotowire.com/basketball/player-stats.php"
urlSalaryNBA="http://www.spotrac.com/nba/rankings/"
stats,players=getRWNBADataPlayers(urlNBA,0)
stats=getSalaryNBADataPlayers(urlSalaryNBA,stats, players)
exportCSV(stats, 0)
stats, players=getRWNBADataPlayers(urlWNBA,1)
stats=getSalaryWNBADataPlayers(stats)
exportCSV(stats, 1)
```

La función getRWNBADataPlayers(url, gender) crea una matriz de listas. Cada lista corresponde a las estadísticas de un jugador o jugadora. Al usar para ambas ligas la misma fuente de datos, que es Rotowire.com el código de *scrapping* es el mismo.

```
def getRWNBADataPlayers(url,gender):
 try:
    if gender==0:
        page = requests.post(url, data={'stat':'Per
Game','season':'2016','submit':'Show
Stats','dpstartwithrange':'10/25/2016','dpendwithrange':'04/13/2017'})
    else:
        page = requests.get(url)
    tree = html.fromstring(page.content)
 except HTTPError as e:
     return
 try:
    tempPlayers = tree.xpath('//tbody/tr/td/a/text()')
    stats=[]
    row=[]
    players={}
    rowStats = tree.xpath('//table/tbody/tr/td/text()')
    for i in range(0, int(len(rowStats)/21)):
        players[tempPlayers[i]]=i
        row.append(tempPlayers[i])
        if isfloat(rowStats[i*21+2]):
            games=int(float(rowStats[i*21+2]))
        row.append(games) #games
        if isfloat(rowStats[i*21+3]):
            row.append(int(float(rowStats[i*21+3])*games)) #minutes
        if isfloat(rowStats[i*21+4]):
            row.append(int(float(rowStats[i*21+4])*games)) #points
        if isfloat(rowStats[i*21+5]):
            row.append(int(float(rowStats[i*21+5])*games)) #rebounds
        if isfloat(rowStats[i*21+6]):
            row.append(int(float(rowStats[i*21+6])*games)) #assists
        if isfloat(rowStats[i*21+7]):
            row.append(int(float(rowStats[i*21+7]))) #steals
        if isfloat(rowStats[i*21+8]):
            row.append(int(float(rowStats[i*21+8])*games)) #blocks
        row.append(0) #sallary
```

```
        row.append(0.0) #sallary/points
        row.append(0.0) #sallary/rebounds
        row.append(0.0) #sallary/assists
        row.append(0.0) #sallary/steals
        row.append(0.0) #sallary/blocks
        stats.append(row)
        row=[]
 except Exception as e:
     print('Error gLDP on line {}'.format(sys.exc_info()[-1].tb_lineno),
type(e).__name__, e)
     return stats, players
 return stats, players
```

La función getSalaryNBADataPlayers(url, stats, player) obtiene los salarios de los jugadores mediante *scrapping* a la página Spotrac.com. Luego introduce los datos de los salarios en la matriz de estadísticas y realiza los cálculos de los atributos relacionados con el salario.

```
   def getSalaryNBADataPlayers(url,stats, players):
 try:
    driver = webdriver.PhantomJS()
    driver.set_window_size(1120, 550)
    driver.get(url)
    driver.maximize_window()

    wait = WebDriverWait(driver, 10)
    wait.until(EC.visibility_of_element_located((By.CLASS_NAME, "tablesorter-
headerRow")))

    content = driver.page_source
    tree = html.fromstring(content)
 except HTTPError as e:
     return
 try:
    names=tree.xpath('//td[@class="rank-name player
noborderright"]/h3/a/text()')
    salaries = tree.xpath('//tbody/tr/td/span[@class="info"]/text()')
 except Exception as e:
    print('Error gLDP on line {}'.format(sys.exc_info()[-1].tb_lineno),
type(e).__name__, e)

 for i in range(0,len(salaries)):
     try:
         salary=int(salaries[i].replace(",","").replace("$",""))
         player=players[names[i]]
         stats[player][8]=salary
         if stats[player][3]!=0:
             stats[player][9]=int(salary/stats[player][3])
         else:
             stats[player][9]="N/A"
         if stats[player][4]!=0:
             stats[player][10]=int(salary/stats[player][4])
         else:
             stats[player][10]="N/A"
         if stats[player][5]!=0:
             stats[player][11]=int(salary/stats[player][5])
         else:
```

```
            stats[player][11]="N/A"
        if stats[player][6]!=0:
            stats[player][12]=int(salary/stats[player][6])
        else:
            stats[player][12]="N/A"
        if stats[player][7]!=0:
            stats[player][13]=int(salary/stats[player][7])
        else:
            stats[player][13]="N/A"
    except Exception as e:
        print('Error gLDP on line {}'.format(sys.exc_info()[-1].tb_lineno),
type(e).__name__, e)
 return stats
```

DATASET

El código crea dos dataset en formato CSV. Estos se pueden visualizar desde la página de Github:

- Dataset de jugadores de la NBA. Podemos ver sus estadísticas básicas, su salario, la relación entre el salario y las estadísticas básicas:

  https://github.com/Bengis/nba-wnba-salary-gap/blob/master/data/nba-stats_out.csv

- Dataset de jugadoras de la WNBA. Podemos ver sus estadísticas básicas, su salario, la relación entre el salario y las estadísticas básicas:

  https://github.com/Bengis/nba-wnba-salary-gap/blob/master/data/wnba-stats_out.csv

Si observamos el dataset podemos ver las enormes discrepancias que existen en el salario. Por ejemplo, la lista de los jugadores y jugadoras con mejor rendimiento de salario por puntos realizados:

Jugador	Salario/Puntos	Jugadora	Salario/Puntos
Nikola Jokic	1207	Tina Charles	156
Derrick Rose	1277	Breanna Stewart	160
Dwayne Wade	1340	Sylvia Fowles	163
Devin Booker	1346	Nneka Ogwumike	164
Sean Kilpatrick	1662	Skylar Diggins-Smith	166

En este caso la brecha salarial solo tiene un orden de magnitud. Pero si tuviésemos en cuenta factores como puntos/partido en lugar de puntos totales los jugadores de la NBA suben a 10000 dólares/punto.

BIBLIOGRAFÍA

- **Various et al.** (2017) XPath Tutorial [en línea] W3Schools.com. https://www.w3schools.com/xml/xpath_intro.asp [Fecha de consulta 20 de octubre de 2017]
- **Various et al.** (2017) Lxml: XML and HTML with Python[en línea] Lxml.de http://lxml.de/ [Fecha de consulta 20 de octubre de 2017]